Spectral Clustering in Heterogeneous Information Networks

Xiang Li,[1] Ben Kao,[2] Zhaochun Ren,[1] Dawei Yin[1]
[1]Data Science Lab, JD.com, China
[2]Department of Computer Science, The University of Hong Kong, Hong Kong
{lixiang81, renzhaochun}@jd.com, kao@cs.hku.hk, yindawei@acm.org

Abstract

A heterogeneous information network (HIN) is one whose objects are of different types and links between objects could model different object relations. We study how spectral clustering can be effectively applied to HINs. In particular, we focus on how meta-path relations are used to construct an effective similarity matrix based on which spectral clustering is done. We formulate the similarity matrix construction as an optimization problem and propose the SClump algorithm for solving the problem. We conduct extensive experiments comparing SClump with other state-of-the-art clustering algorithms on HINs. Our results show that SClump outperforms the competitors over a range of datasets w.r.t. different clustering quality measures.

1 Introduction

A *heterogeneous information network* (HIN) is a network structure whose objects could assume different object types and links between objects could represent different kinds of relationships between objects. HINs are ubiquitous and are used to model many different kinds of real-world data. For example, the Facebook open graph[1] models *users*, *posts*, *events*, and *pages* as four different kinds of objects. A user can *publish* a post, *attend* an event, or *like* a page, which illustrates three different kinds of connections relating a user object to a post/event/page object. Compared with homogeneous networks (in which objects are of single type and links model single relation), HINs are a richer construct for capturing complex objects and their relations.

Data analytics on HINs has been an active area of research (Sun et al. 2011; Li et al. 2016). Being a fundamental task in machine learning and data mining, cluster analysis has found interesting applications in HINs. For example, clustering Facebook users based on their interests enables effective target and viral marketing (Li et al. 2017). Even though spectral clustering is very effective for data that is modeled as (homogeneous) network/graph (Liu et al. 2013), there are surprisingly few studies that apply spectral clustering to HINs. The objective of this paper is to study how spectral clustering can be effectively applied to HINs to improve clustering quality.

[1]https://developers.facebook.com/docs/sharing/opengraph

Spectral clustering transforms clustering into a graph partitioning problem that optimizes a certain criterion that measures the quality of the partitions, such as the *normalized cuts* (Shi and Malik 2000). Generally, given a set of objects $X = \{x_1, x_2, ..., x_n\}$, standard spectral clustering methods first construct an undirected graph $G = (X, S)$, where X denotes the vertex set and S is a matrix such that S_{ij} measures the similarity between objects x_i and x_j.[2] Then, the Laplacian matrix L_S is computed based on which eigen-decomposition is performed to obtain k eigenvectors that correspond to the k smallest eigenvalues, where k is the number of desired clusters. These eigenvectors are used as new feature space of objects. Finally, a post-processing step, such as k-means (Ng, Jordan, and Weiss 2002) and spectral rotation (Yu and Shi 2003) is applied to partition the objects into k clusters.

Previous studies have shown that the performance of spectral clustering highly depends on the "quality" of the similarity matrix (Nie et al. 2016). Intuitively, a high-quality matrix S is one such that S_{ij} is large if objects x_i and x_j ought to be in the same cluster and S_{ij} is small otherwise. The challenges of constructing a high-quality matrix S in HINs are two-fold. First, although the similarity between two objects in an HIN can be measured by conventional network distances (such as shortest paths or random-walk based similarity), previous works have shown that *meta-path/meta-structure based similarity* is much more effective in HINs (Sun et al. 2011; Huang et al. 2016; Fang et al. 2016). (We use meta-paths in this paper and our method can be easily adapted to using meta-structures.) A meta-path is a sequence of object types that expresses a path-based relation between two objects. For example, in Facebook, the meta-path *User-Event-User* represents the relation between users who have attended the same event; the meta-path *User-Page-User* captures the relation between two users who have *liked* the same product page. An interesting issue is how various meta-paths can be integrated to formulate a similarity matrix that exhibits a clear clustering structure. Second, a theoretically infinite number of meta-paths can be derived from an HIN (with meta-paths composed of different object types and of various lengths).

[2]Given a matrix M, we use M_{ij} or $M[i, j]$ to refer to the (i,j)-th entry of M.

However, generally, only a small subset of them are useful for a given clustering task. For example, the meta-path *User-Page-User* is useful for clustering users based on their interests. The same meta-path, however, is much less useful if we want to cluster users based on their geographic locations. A mechanism for weighing the relative importance of meta-paths is thus essential.

We propose the SClump algorithm, which stands for Spectral Clustering Using Meta-Paths, to address the above problems. SClump uses meta-paths to construct the similarity matrix S. The matrix S is refined through an iterative process, whose goal is to optimize an objective function that captures the quality of S. During the process, weights of meta-paths are also learned. Here, we summarize our contributions.

• We show how spectral clustering can be effectively applied to HINs. In particular, we show how meta-paths are used to construct an effective similarity matrix.

• We propose the SClump algorithm, which employs an iterative learning process via which the similarity matrix and weights of meta-paths are mutually refined.

• We conduct extensive experiments on real datasets to show the effectiveness of SClump. We compare SClump with the state-of-the-art clustering methods for HINs. Our results show that spectral clustering is an effective approach for HINs and that SClump significantly outperforms existing methods.

The rest of the paper is organized as follows. Section 2 summarizes related works on general spectral clustering and existing clustering methods for HINs. Section 3 gives formal definitions of related concepts and the problem we study. Section 4 describes our algorithm SClump. Section 5 gives the experimental results and Section 6 concludes the paper.

2 Related Work

The problem of clustering objects (vertices) in HINs has been an interesting area of research. Both unsupervised methods (e.g., (Zhang et al. 2016)) and semi-supervised methods (e.g., (Li et al. 2017)) have been proposed. Most of these methods measure objects' similarity based on the links connecting them (e.g., (Zhou and Liu 2013)). There are also methods that consider both links and objects' attribute values (e.g., (Sun, Aggarwal, and Han 2012)).

As a graph-based clustering method, general spectral clustering is a well studied subject (Shi and Malik 2000; Zelnik-Manor and Perona 2005; Huang et al. 2009; Li et al. 2018). Some of these studies focus on the computational efficiency (Song et al. 2008; Chen and Cai 2011), while others focus on the probabilistic interpretation of the method (Meila and Shi 2001; Nadler et al. 2005). Spectral clustering has been shown to be very effective for unstructured data (in which objects are associated with fixed-length feature vectors) (Ng, Jordan, and Weiss 2002) and homogeneous network data (Liu et al. 2013). However, very few works on spectral clustering have been carried out in the context of HINs. In the following, we briefly describe two representative works, namely, SRC (Long et al. 2006) and Het-RSC (Sengupta and Chen 2015).

SRC is a spectral clustering method for clustering multi-type objects. Given an HIN, SRC first derives a relation table (or a matrix) for each type of link in the network. These relation matrices are then collectively factorized through an iterative process to obtain low-dimensional embeddings of the objects. These embeddings serve as objects' feature vectors on which k-means is applied to cluster the objects.

Het-RSC is a regularized spectral clustering method for HINs. Given an HIN with n objects of m types, Het-RSC first constructs an adjacency matrix $A \in \mathbb{R}^{n \times n}$ such that A_{ij} gives the number of links between objects x_i and x_j. Then, eigen-decomposition is performed on the regularized graph Laplacian of A. Let $E = [\boldsymbol{e}_1; \boldsymbol{e}_2; ...; \boldsymbol{e}_{mk}]$ denote the mk eigenvectors that correspond to the mk eigenvalues of the largest absolute values. For each object type T, the algorithm extracts the rows in E that correspond to the objects of type T. These rows form those objects' feature vectors. k-means is then applied to cluster the objects based on the vectors.

While SRC and Het-RSC do not make use of meta-paths, there are (non-spectral-clustering) meta-path-based methods. Two representative ones are PathSelClus (Sun et al. 2012) and HMFClus-S (Zhang et al. 2016). PathSelClus uses a probabilistic generative model for clustering. Specifically, each meta-path $\mathcal{P}$ derives a relation matrix $M_{\mathcal{P}}$ such that $M_{\mathcal{P}}[i, j]$ records the number of instances of $\mathcal{P}$ that relate objects x_i and x_j. Given a set of meta-paths, their corresponding relation matrices are taken as *evidence* based on which a probabilistic model of hidden clusters is derived.

HMFClus-S (Zhang et al. 2016) is a state-of-the-art clustering algorithm for HINs. The method applies nonnegative matrix factorization to the relation matrices that are derived from meta-paths. These matrices produce latent factors (low dimensional embedding vectors) for objects from which a consensus is learned. A meta-path based similarity regularization step is further employed. A post-processing step (e.g., k-means) using the low dimensional latent factors is applied to cluster objects.

Recently, network embedding has emerged as a useful tool to mine networked data. The idea is to learn low dimensional embedding vectors to represent objects in a network (Tang et al. 2015; Grover and Leskovec 2016; Perozzi, Al-Rfou, and Skiena 2014; Kipf and Welling 2016; Hamilton, Ying, and Leskovec 2017). The embedding vectors can then be used in various data analysis tasks, such as clustering, classification, and similarity search. Recent works on embedding objects in HINs include (Fu, Lee, and Lei 2017; Dong, Chawla, and Swami 2017; Tang, Qu, and Mei 2015; Shi et al. 2018; Chang et al. 2015).

HIN2Vec (Fu, Lee, and Lei 2017) uses meta-paths for embedding HINs. They construct a binary classifier model that predicts, given a pair of objects in an HIN, whether a meta-path-based relationship should exist between the objects. Taking embedding vectors of objects as parameters, the model is trained by maximizing the likelihood of the given training data. By performing random walk from one object node to another, a positive training sample that connects the two nodes with the meta-path derived from the random walk path is obtained; Negative samples are drawn by negative sampling.

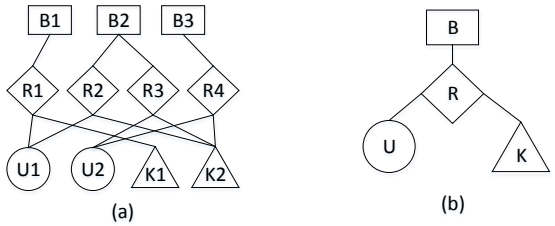


Figure 1: An HIN (a) and its schematic graph (b)

metapath2vec (Dong, Chawla, and Swami 2017) is a meta-path-based embedding method for HINs, which performs meta-path-based random walks to construct a heterogeneous neighborhood of a node. metapath2vec++ is an extension to metapath2vec, which uses node type information in distinguishing context nodes. Cluster analysis can be done on the embedding vectors (obtained from HIN2Vec or metapath2vec++) by standard methods such as *k*-means.

3 Definitions

In this section we give a formal problem definition.

Definition 1 ***Heterogeneous Information Network (HIN).*** *Let $\mathcal{T} = \{T_1, ..., T_m\}$ be a set of m object types. For each type T_i, let $\mathcal{X}_i$ be the set of objects of type T_i. An HIN is a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \bigcup_{i=1}^{m} \mathcal{X}_i$ is a set of nodes and $\mathcal{E}$ is a set of links, each represents a binary relation between two objects in $\mathcal{V}$.* □

Definition 2 ***Network schema.*** *A network schema is a meta template of an HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Let (1) $\phi : \mathcal{V} \to \mathcal{T}$ be an object-type mapping that maps an object in $\mathcal{V}$ into its type, and (2) $\psi : \mathcal{E} \to \mathcal{R}$ be a link-relation mapping that maps a link in $\mathcal{E}$ into a relation in a set of relations $\mathcal{R}$. The network schema of an HIN $\mathcal{G}$, denoted by $T_{\mathcal{G}} = (\mathcal{T}, \mathcal{R})$, shows how objects of different types are related by the relations in $\mathcal{R}$. A schematic graph is used to represent $T_{\mathcal{G}}$ with $\mathcal{T}$ and $\mathcal{R}$ being the node set and the edge set, respectively. Specifically, there is an edge (T_i, T_j) in the schematic graph iff there is a relation in $\mathcal{R}$ that relates objects of type T_i to objects of type T_j.* □

Example: Figure 1(a) shows a Yelp network that includes four object types: $\mathcal{T}$ = {review(♢), business(□), user(○), keyword(△)}. The set $\mathcal{R}$ consists of three relations, which are illustrated by the three edges in the schematic graph (Figure 1(b)). For example, the edge B–R in Figure 1(b) could represent the relation that a (B)usiness is given a (R)eview; the edge R–K could indicate that a (K)eyword is mentioned in a (R)eview.

Definition 3 ***Meta-path.*** *A meta-path $\mathcal{P}$ is a path defined on the schematic graph of a network schema. A meta-path $\mathcal{P}$: $T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_l} T_{l+1}$ defines a composite relation $R = R_1 \circ \cdots \circ R_l$ that relates objects of type T_1 to objects of type T_{l+1}. If two objects x_u and x_v are related by the composite relation R, then there is a path, denoted by $p_{x_u \rightsquigarrow x_v}$, that connects x_u to x_v in $\mathcal{G}$. Moreover, the sequence of links in $p_{x_u \rightsquigarrow x_v}$ matches the sequence of relations R_1, ..., R_l based on the link-relation mapping ψ. We say that $p_{x_u \rightsquigarrow x_v}$ is a path instance of $\mathcal{P}$, denoted by $p_{x_u \rightsquigarrow x_v} \vdash \mathcal{P}$.* □

For example, the path $p_{B1 \rightsquigarrow B2}$ = B1 → R1 → U1 → R2 → B2 in Figure 1(a) is an instance of the meta-path Business-Review-User-Review-Business (abbrev. BRURB) that captures the relation between two businesses that have been reviewed by the same customer; the path $p_{B2 \rightsquigarrow B3}$ = B2 → R3 → K2 → R4 → B3 is an instance of the meta-path Business-Review-Keyword-Review-Business (abbrev. BRKRB) that captures the relation between two businesses that have reviews containing the same keyword.

Definition 4 ***Clustering in HINs.*** *Given an HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a target object type T_h, the number of clusters k, and a set of meta-paths $\mathcal{PS}$, the problem of HIN clustering is to partition the objects in $\mathcal{X}_h$ into k disjoint clusters $\mathcal{C} = \{C_1, ..., C_k\}$.* □

4 Algorithm

In this section we describe our algorithm SClump.

Similarity Matrix

The key step in spectral clustering is to construct a high-quality similarity matrix S. For HINs, meta-paths have been effectively used to measure object similarity. For example, given a meta-path $\mathcal{P}$, *PathSim* (Sun et al. 2011) measures the similarity between two objects x_u and x_v w.r.t. $\mathcal{P}$ by counting the number of path instances of $\mathcal{P}$ that connect the two objects. Specifically, we have,

$$S_{\mathcal{P}}(x_u, x_v) = \frac{2 \times |\{p_{x_u \rightsquigarrow x_v} : p_{x_u \rightsquigarrow x_v} \vdash \mathcal{P}\}|}{|\{p_{x_u \rightsquigarrow x_u} : p_{x_u \rightsquigarrow x_u} \vdash \mathcal{P}\}| + |\{p_{x_v \rightsquigarrow x_v} : p_{x_v \rightsquigarrow x_v} \vdash \mathcal{P}\}|}. \quad (1)$$

Given a set of meta-paths $\mathcal{PS}$, each meta-path $\mathcal{P}_i \in \mathcal{PS}$ derives a similarity matrix $S_{\mathcal{P}_i}$ based on Equation 1. We construct a matrix W as the weighted sum of the matrices:

$$W = \sum_{i=1}^{|\mathcal{PS}|} \lambda_i S_{\mathcal{P}_i}. \quad (2)$$

Our objective is to utilize the information provided by W to derive a similarity matrix S that exhibits a *clear clustering structure*. Intuitively, S has a clustering structure if it mimics a block-diagonal matrix (under certain matrix permutation). Figure 2(b) illustrates such a structure. Each block in the matrix indicates the memberships of a target cluster. Given the set of objects $\mathcal{X}_h$ of a target type T_h, we construct a graph G based on S. Specifically, $\forall x_u, x_v \in \mathcal{X}_h$, x_u and x_v are connected by an edge with weight S_{uv} iff $S_{uv} > 0$. The purpose is to use the meta-path based similarities (i.e., W and hence S) as a better measure of object similarity over the straightforward direct linkage between objects given by the original HIN. If S is nonnegative, the graph Laplacian $L_S = D - (S + S^T)/2$, where D is a diagonal matrix with $D_{ii} = \sum_j (S_{ij} + S_{ji})/2$, has the following property (Marsden 2013; Chung 1997):

Theorem 1 *A graph G has k connected components iff the multiplicity of eigenvalue 0 in the graph Laplacian L_S is k.*

Theorem 1 implies that if $S \in \mathbb{R}^{n\times n}$ and $rank(L_S) = n - k$, where n is the number of objects, the corresponding graph G will have exactly k connected components. Based on spectral graph theory, each component can be considered a cluster. Let $\boldsymbol{\lambda}$ be a vector that represents the meta-paths' weights (λ_i's). We determine the matrix S and meta-path weights $\boldsymbol{\lambda}$ by solving an optimization problem with the objective function $\Gamma(S, \boldsymbol{\lambda})$ given below:

$$\text{minimize } \Gamma(S, \boldsymbol{\lambda}) = ||S - W||_F^2 + \alpha||S||_F^2 + \beta||\boldsymbol{\lambda}||_2^2,$$
$$s.t.\ \sum_{j=1}^{n} S_{ij} = 1;\ S_{ij} \geq 0;\ \sum_{i=1}^{|\mathcal{PS}|} \lambda_i = 1;\ \lambda_i \geq 0;\ rank(L_S) = n - k.$$

The objective function consists of three terms. The first term $||S - W||_F^2$ learns S that best approximates W. To prevent overfitting, we introduce two regularization terms $||S||_F^2$ and $||\boldsymbol{\lambda}||_2^2$. The first four constraints normalize the entries of S and the meta-path weights, and to prevent rows of S having all 0 values. The constraint $rank(L_S) = n$ - k is used to force k connected components in S. Our aim is to learn a similarity matrix S with exactly k connected components from a set of meta-path based similarity matrices that are summarized in W.

Optimization

The optimization problem is non-convex due to the constraint $rank(L_S) = n - k$. It is thus difficult to optimize the problem directly. We transform the original problem to:

$$\min\ ||S - \sum_{i=1}^{|\mathcal{PS}|} \lambda_i S_{\mathcal{P}_i}||_F^2 + \alpha||S||_F^2 + \beta||\boldsymbol{\lambda}||_2^2 + 2\gamma \sum_{i=1}^{k} \sigma_i(L_S),$$
$$s.t.\ \sum_{j=1}^{n} S_{ij} = 1,\ S_{ij} \geq 0,\ \sum_{i=1}^{|\mathcal{PS}|} \lambda_i = 1,\ \lambda_i \geq 0, \tag{3}$$

where $\sigma_i(L_S)$ denotes the i-th smallest eigenvalue of L_S. Since L_S is semi-definite, $\sigma_i(L_S) \geq 0$. By setting a large γ value, we force the term $\sum_{i=1}^{k} \sigma_i(L_S)$ to zero to guarantee $rank(L_S) = n - k$. According to the Ky-Fan Theorem (Fan 1949), we have,

$$\sum_{i=1}^{k} \sigma_i(L_S) = \min_{F \in \mathbb{R}^{n\times k}, F^T F = I} tr(F^T L_S F), \tag{4}$$

where $tr(\cdot)$ is the trace operator. The optimization problem is thus equivalent to:

$$\min\ ||S - \sum_{i=1}^{|\mathcal{PS}|} \lambda_i S_{\mathcal{P}_i}||_F^2 + \alpha||S||_F^2 + \beta||\boldsymbol{\lambda}||_2^2 + 2\gamma\, tr(F^T L_S F),$$
$$s.t.\ \sum_{j=1}^{n} S_{ij} = 1,\ S_{ij} \geq 0,\ \sum_{i=1}^{|\mathcal{PS}|} \lambda_i = 1,\ \lambda_i \geq 0,\ F \in \mathbb{R}^{n\times k},\ F^T F = I, \tag{5}$$

where S, $\boldsymbol{\lambda}$ and F are variables. SClump solves Problem (5) using an iterative update approach. In each iteration, two of the above three variables are fixed, while the remaining one is updated. We now describe the update procedure.

[Update F with S and $\boldsymbol{\lambda}$ fixed]. With fixed S and $\boldsymbol{\lambda}$, Problem (5) can be simplified as:

$$\min_{F \in \mathbb{R}^{n\times k},\ F^T F = I} tr(F^T L_S F). \tag{6}$$

According to the Ky-Fan Theorem, Problem (6) has a closed-form solution that corresponds to the subspace spanned by the k eigenvectors with the k smallest eigenvalues of L_S.

[Update S with F and $\boldsymbol{\lambda}$ fixed]. We extract terms and constraints that are relevant to S from Problem (5) and derive:

$$\min_{\sum_{j=1}^{n} S_{ij}=1,\ S_{ij}\geq 0} ||S - W||_F^2 + \alpha||S||_F^2 + 2\gamma\, tr(F^T L_S F). \tag{7}$$

Since F is fixed, Problem (7) can be rewritten as

$$\min_{\sum_{j=1}^{n} S_{ij}=1,\ S_{ij}\geq 0} ||S-W||_F^2 + \alpha||S||_F^2 + \gamma \sum_{i,j} ||\vec{f}_i - \vec{f}_j||_2^2 S_{ij}, \tag{8}$$

where $\vec{f}_i$ and $\vec{f}_j$ are the i-th and the j-th row vectors in F, respectively. We can decompose Equation (8) into a number of subproblems, each corresponds to an object x_i:

$$\min_{\vec{s}_i \mathbf{1}=1, \vec{s}_i \geq \mathbf{0}} ||\vec{s}_i - \vec{w}_i||_2^2 + \alpha||\vec{s}_i||_2^2 + \gamma \sum_{j} ||\vec{f}_i - \vec{f}_j||_2^2 S_{ij}, \tag{9}$$

where $\vec{s}_i$ and $\vec{w}_i$ are the i-th row of S and W, respectively. Now, rewrite Problem (9) as

$$\min_{\vec{s}_i \mathbf{1}=1, \vec{s}_i \geq \mathbf{0}} ||\vec{s}_i - \vec{p}_i||_2^2, \tag{10}$$

where $\vec{p}_i$ is the i-th row of $P = (2W - \gamma Q)/(2 + 2\alpha)$ and $Q \in \mathbb{R}^{n\times n}$ with $Q_{ij} = ||\vec{f}_i - \vec{f}_j||_2^2$. Problem (10) is convex and it can be solved by an efficient iterative algorithm (Huang, Nie, and Huang 2015).

[Update $\boldsymbol{\lambda}$ with F and S fixed]. We rewrite Problem (5) as

$$\min_{\sum_{i=1}^{|\mathcal{PS}|} \lambda_i = 1,\ \lambda_i \geq 0} ||S - \sum_{i=1}^{|\mathcal{PS}|} \lambda_i S_{\mathcal{P}_i}||_F^2 + \beta||\boldsymbol{\lambda}||_2^2, \tag{11}$$

which is a quadratic programming problem that can be solved by (Boyd and Vandenberghe 2004).

Finally, SClump constructs the graph G from S as explained earlier and determines the k connected components in G. These components form the desired clusters. Algorithm 1 shows the pseudo code of SClump. The major computation performed by SClump is the iterative updates of F and S. Their complexities are $O(kn^2)$ and $O(n^2)$, respectively, where n is the number of objects and k is the number of clusters. Since typical matrices we deal with are sparse, the time complexity of updating F can be reduced to $O(knt)$, where t is the average number of non-zero entries per row in the matrix. The convergence of SClump can also be proved — Our update process follows a general coordinate descent approach. In particular, each update in Equations. (6), (7) and (11) decreases the value of the objective function in Problem (5). Since the objective function is lower bounded by 0, the convergence of SClump is guaranteed.

5 Experiment

In this section we evaluate the performance of SClump. We use three popular measures, namely, *normalized mutual information (NMI)*, *purity*, and *rand index (RI)*, to evaluate clustering quality (Lin and Cohen 2010). Note that values of all three measures range from 0 to 1, with a larger value indicating a better clustering quality.

Algorithm 1 SClump

Input: $\mathcal{G}, T_h, k, \mathcal{PS}$.
Output: $S, \boldsymbol{\lambda}, \mathcal{C}$

Calculate $S_{\mathcal{P}}$ for each $\mathcal{P} \in \mathcal{PS}$
Normalize $S_{\mathcal{P}}$ by setting $\sum_{j=1}^{n} S_{\mathcal{P}}[i,j] = 1$
Initialize $\boldsymbol{\lambda}$, $W^{(0)}$ and $t = 1$
while not convergence **do**
    Update F by calculating the k eigenvecters of L_S corresponding to the k smallest eigenvalues
    for $i \leftarrow 1$ to n **do**
        Update $\vec{s}_i$ by solving Problem (10)
    end for
    Update $\boldsymbol{\lambda}$ by solving Problem (11)
    Update $W^{(t)}$ by Eq. (2)
    $t = t + 1$
end while
Identify each connected component C_r in S as a cluster and obtain clusters $\mathcal{C} = \{C_r\}_{r=1}^{k}$
return $\mathcal{C} = \{C_1, ..., C_k\}$

Clustering Tasks

We use three datasets *Freebase*, *DBLP* and *Yelp* in the experiments. Freebase is a knowledge base that models entities and their relationships as a graph. DBLP is a bibliographic network of scientific publications. Yelp is a business referral service, whose data includes various information of businesses such as customer reviews. From these datasets, we formulated four clustering tasks:

• **TV-Series**. We extracted an HIN related to TV series from Freebase. This HIN includes objects of three types: 191 TV series (S), 19 directors (D) and 26 creators (C). There are two types of links: D-S (director of a TV series) and C-S (creator of a TV series). We take Series (S) as the target type and the clustering task is to cluster Series objects based on their genres. (The three hidden genres are *comedy drama*, *soap opera*, and *police procedural*.) We use two meta-paths, namely, SCS (two series that are created by the same creator) and SDS (two series that are directed by the same director).

• **DBLP**. We extracted an HIN consisting of 2,591 authors (A), 4,269 papers (P), 20 publication venues (V) and 3,219 paper keywords (T). These authors focus on different (hidden) research areas: *data mining* (DM), *database* (DB), *information retrieval* (IR) and *artificial intelligence* (AI). Three relations exist between these objects: P-A (authorship), P-V (paper published in a venue), P-T (paper contains a keyword). We consider four meta-paths: {APA, APAPA, APVPA, APTPA}. The clustering task is to cluster authors (target type) based on their research areas.

• **Yelp-B**. We extracted information from Yelp to construct an HIN with 1,448 businesses (B), 40 cities (C), 6,577 users (U) and 354 business category objects (A). These businesses are located in four (hidden) regions: *North Carolina* (NC), *Wisconsin* (WI), *Pennsylvania* (PA) and *Edinburgh, UK* (EDH). Objects are linked by three relations: B-C (business in a city), U-B (customer of a business), B-A (business of a category). We use three meta-paths: {BCB, BUB, BAB}. The clustering task is to cluster businesses (target type) based on the regions they reside.

• **Yelp-R**. We extracted an HIN that includes 2,224 restaurants (B), 16,020 reviews (R), 614 users (U) and 92 food relevant keywords (K). These restaurants provide different food: *Italian*, *Chinese* and *Japanese*. Links include R-B (review for a restaurant), U-R (customer of a restaurant) and K-R (keyword in a review). We use meta-paths {BRURB, BRKRB} to cluster restaurants (target type) based on food categories.

Clustering Quality

We compare SClump with 7 other algorithms, which can be categorized into four groups: (1) SRC and Het-RSC, which are spectral clustering methods for HINs that do not use meta-paths. (2) PathSelClus and HMFClus-S, which are non-spectral-clustering methods that use meta-paths. (3) HIN2Vec and metapath2vec++, which are state-of-the-art network embedding methods for HINs that use meta-paths. We apply k-means to cluster objects based on their low-dimensional embedding vectors. (4) A variant of SClump. Note that SClump uses meta-paths to derive similarity between objects. To understand the effectiveness of the meta-path-based similarity measure in solving the HIN clustering problem, we consider *random walk with restart* (RWR) (Tong, Faloutsos, and Pan 2006) as an alternative measure. We call this variant SClump-RWR.

In the experiments, parameters of the methods are set to the values reported in their original papers. We use k-means as the final post-processing step to return clusters. For this step, we run k-means 10 times with random centroids and the most frequent cluster assignment is reported. The only exception is PathSelClus, which uses a probabilistic generative model for clustering. For metapath2vec++, each meta-path independently induces a low dimensional embedding vector for an object. We assign meta-paths equal weights and calculate the final embedding vector for each object by taking the weighted average. For SClump, we set $\alpha = 0.1$, $\beta = 10$ for Yelp-R and $\alpha = 0.5$, $\beta = 10$ for other clustering tasks. Moreover, γ is set according to (Nie et al. 2016).

Tables 1, 2 and 3 show the clustering quality of the algorithms using NMI, purity and RI as quality measures, respectively. We use PSC, HMF and mp2vec++ as shorthands for PathSelClus, HMFClus-S and metapath2vec++. We compare the eight algorithms for the four clustering tasks under three quality measures. There are in total 12 (4 tasks $\times$ 3 measures) contests. Each row in the tables corresponds to a (task-measure) contest with the winner's score shown in boldface. We summarize our observations as follows.

• SRC and Het-RSC are spectral clustering methods that do not use meta-paths. Comparing them against SClump (which uses both spectral clustering and meta-paths), we see that SClump consistently provides high clustering quality while SRC and Het-RSC perform well in some cases but poorly in others. For example, for the (DBLP-NMI) contest, SClump (0.6917) significantly outperforms SRC (0.4826); while for the (Yelp-B-NMI) contest, SClump (1.0) is perfect but Het-RSC (0.4908) is contest-worst. These results show

Datasets	SRC	Het-RSC	PSC	HMF	HIN2Vec	mp2vec++	SClump-RWR	SClump
Series	0.5688	0.5367	0.5239	0.5198	**0.5822**	0.5330	0.5607	0.5787
DBLP	0.4826	0.6789	0.6770	0.5698	0.2555	0.4122	0.0076	**0.6917**
Yelp-B	0.9671	0.4908	0.8891	0.7044	0.5303	0.9964	0.8997	**1.000**
Yelp-R	0.7378	0.6006	0.7472	0.7177	0.6170	0.3903	0.0019	**0.7836**

Table 1: Clustering results (NMI) on the four clustering tasks

Datasets	SRC	Het-RSC	PSC	HMF	HIN2Vec	mp2vec++	SClump-RWR	SClump
Series	0.7958	0.8205	0.7962	0.8057	0.7853	0.7749	0.8429	**0.8482**
DBLP	0.7058	**0.8854**	0.8725	0.7885	0.5461	0.7445	0.3084	0.8838
Yelp-B	0.9917	0.6691	0.9316	0.7691	0.6802	0.9993	0.9675	**1.000**
Yelp-R	0.9141	0.7879	0.9172	0.8543	0.8147	0.6560	0.4353	**0.9308**

Table 2: Clustering results (Purity) on the four clustering tasks

Datasets	SRC	Het-RSC	PSC	HMF	HIN2Vec	mp2vec++	SClump-RWR	SClump
Series	**0.8229**	0.7979	0.7983	0.7866	0.7892	0.7876	0.7999	0.8008
DBLP	0.7701	**0.8975**	0.8891	0.8440	0.6749	0.7949	0.2604	0.8959
Yelp-B	0.9907	0.7464	0.9453	0.8333	0.7559	0.9992	0.9638	**1.000**
Yelp-R	0.9088	0.7876	0.9110	0.8783	0.8476	0.7182	0.3547	**0.9252**

Table 3: Clustering results (RI) on the four clustering tasks

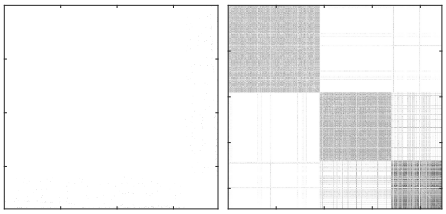
that meta-paths are very useful in improving clustering in HINs.

• PSC and HMF use meta-paths but they are not spectral clustering methods. From the tables, we see that SClump outperforms them over all 12 contests. In particular, for (Yelp-B-Purity), PSC (0.9316) and HMF (0.7691) are significantly outperformed by SClump, which scores a perfect 1. The performance gaps show that spectral clustering is effective in clustering HIN objects.

• HIN2Vec and mp2vec++ are embedding methods for HINs. While HIN2Vec achieves the best performance in the (Series-NMI) contest (in which SClump is a close second), SClump outperforms them over all other contests. Compared with SClump, HIN2Vec and mp2vec++ are non-spectral clustering methods. Moreover, they do not adaptively assign weights to meta-paths. These contribute to their poorer performance.

• SClump-RWR uses random walks rather than meta-paths to measure object similarity in HINs. From the tables, we see that SClump consistently outperforms SClump-RWR over all the contests. This shows that meta-paths are more effective in capturing similarity between objects. This observation is also consistent with some previous works on mining heterogenous information networks (Sun et al. 2011).

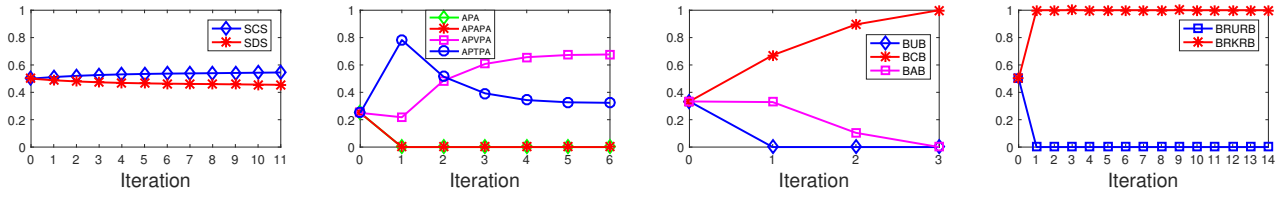
• SClump gives the best overall performance among the eight methods. It wins 8 out of the 12 contests. For the contests that SClump does not win, its performance is very close to that of the winner. For example, for (DBLP-Purity), SClump (0.8838) is very close to the winner Het-RSC (0.8854). Such is also the case for the (DBLP-RI) contest. The results show that SClump's approach of integrating spectral clustering and meta-paths is highly effective for cluster analysis on HINs.



(a) Het-RSC (b) SClump

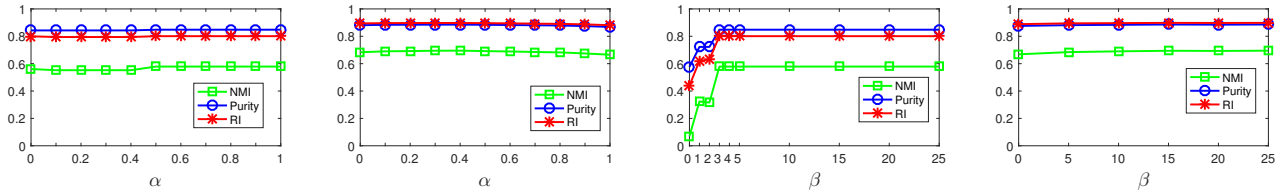
Figure 2: Visualizing the similarity matrices for task Yelp-R

To better understand the advantage of SClump over other spectral clustering methods, let us take a close look at the similarity matrices SClump constructs. As we have discussed in Section 4, a good similarity matrix is crucial for the success of spectral clustering. Intuitively, the similarity matrix S should exhibit a clustering structure visualized as a block-diagonal matrix. To illustrate, Figure 2 compares the similarity matrices constructed by Het-RSC and SClump for the clustering task Yelp-R. Each sub-figure shows a matrix by varying a pixel's darkness according to the corresponding matrix entry value. Objects are re-arranged in the matrix through matrix permutation so that objects assigned to the same cluster are grouped together. Since Het-RSC constructs the matrix based only on the links given in an HIN, the constructed matrix (Figure 2(a)) is very sparse and the pixels are hardly visible. In contrast, SClump uses meta-paths to provide a much richer similarity measure between objects. This results in a more definite clustering structure as evidenced by the more "embossed" blocks shown (Figure 2(b)). This explains the good performance of SClump.



(a) TV-Series (b) DBLP (c) Yelp-B (d) Yelp-R

Figure 3: Meta-path weight learning



(a) TV-Series (b) DBLP (c) TV-Series (d) DBLP

Figure 4: Parameter analysis

Meta-path Weight Learning

SClump uses an iterative update approach to learn meta-path weights $\boldsymbol{\lambda}$ (see Section 4). Figure 3 shows how the weights of the various meta-paths change over the iterations under the four clustering tasks. We see that the weights converge very quickly taking only 1 to 5 iterations. This shows that SClump is practically efficient. We further make the following interesting observations from the figures:

• The task TV-Series is to cluster series by their genres. Figure 3(a) shows that the meta-paths SCS (series with the same creator) and SDS (series with the same director) carry very similar weights. This suggests that both creators and directors are important in determining the genre of their works.

• The task DBLP is to cluster authors by their research areas. In Figure 3(b), we see that the meta-paths APVPA (authors that publish papers in the same venue) and APTPA (authors that publish papers with the same keyword) are given higher weights than APA (co-authorship) and APAPA (2-hop co-authorship). This sounds counter-intuitive because APA (co-authorship), for example, should be highly relevant in determining an author's research area. The reason why APA is given a very low weight by SClump is that APA is a very *sparse* relation. A typical author only co-authors with a handful of others in the research community. Moreover, authors related by APA are necessarily related by APVPA too. SClump correctly selects APVPA and APTPA over APA and APAPA as the more useful meta-paths in the clustering task.

• The task Yelp-B is to cluster businesses by their locations (regions). Figure 3(c) shows that SClump correctly assigns BAB (businesses in the same category), which is irrelevant to the locations of the businesses, a weight of 0. SClump also correctly gives all the weight (1) to BCB (businesses located in the same city), which is the determining feature of location. Note that SClump gives BUB (businesses visited by the same user) a 0 weight. Even though BUB is a relevant relation (due to users locality), the clustering task does not need this relation as the determining relation BCB is picked. This result shows that SClump's weight learning strategy is very effective in feature (meta-paths) selection for clustering.

• The task Yelp-R is to cluster restaurants by the kind of food served. From Figure 3(d), we see that SClump correctly gives BRKRB (restaurants whose reviews contain the same food keyword), which is most relevant to the task, a weight of 1. In contrast, BRURB (restaurants reviewed by the same user) is given a weight of 0 because users could visit restaurants that serve different kinds of food. This again shows the effectiveness of SClump in learning meta-path weights. Another interesting observation is that for Yelp-R, SClump takes only one iteration to identify the correct weighting.

SClump uses two parameters, α and β, to control the regularization terms of S and $\boldsymbol{\lambda}$, respectively. For the tasks TV-Series and DBLP, the (α, β) values chosen were (0.5,10). Figure 4 shows a sensitivity analysis on α and β. From the figure, we see that SClump gives very stable performances over a wide range of parameter values.

6 Conclusions

We studied the problem of clustering objects in an HIN. We proposed the SClump algorithm which takes a spectral clustering approach. Different from existing spectral clustering algorithms for HINs, SClump uses meta-paths in the construction of an effective similarity matrix. Through an iterative learning process, SClump refines the similarity matrix and the meta-paths' weights. Our experimental results show that SClump outperforms existing techniques in terms of a number of clustering quality measures. The superior performance of SClump comes from its ability to construct a similarity matrix that exhibits a clear clustering structure. The iterative process is also efficient, generally taking only a few iterations until convergence.

7 Acknowledgments

This research is supported by Hong Kong RGC General Research Fund 17254016.

References

Boyd, S., and Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

Chang, S.; Han, W.; Tang, J.; Qi, G.-J.; Aggarwal, C. C.; and Huang, T. S. 2015. Heterogeneous network embedding via deep architectures. In *KDD*, 119–128.

Chen, X., and Cai, D. 2011. Large scale spectral clustering with landmark-based representation. In *AAAI*, 313–318.

Chung, F. R. 1997. *Spectral graph theory*. Number 92. American Mathematical Soc.

Dong, Y.; Chawla, N. V.; and Swami, A. 2017. metapath2vec: Scalable representation learning for heterogeneous networks. In *KDD*, 135–144.

Fan, K. 1949. On a theorem of weyl concerning eigenvalues of linear transformations. *PNAS* 35(11):652–655.

Fang, Y.; Lin, W.; Zheng, V. W.; Wu, M.; Chang, K. C.-C.; and Li, X.-L. 2016. Semantic proximity search on graphs with metagraph-based learning. In *ICDE*, 277–288.

Fu, T.-y.; Lee, W.-C.; and Lei, Z. 2017. Hin2vec: Explore meta-paths in heterogeneous information networks for representation learning. In *CIKM*, 1797–1806.

Grover, A., and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *KDD*, 855–864.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*.

Huang, L.; Yan, D.; Taft, N.; and Jordan, M. I. 2009. Spectral clustering with perturbed data. In *NIPS*, 705–712.

Huang, Z.; Zheng, Y.; Cheng, R.; Sun, Y.; Mamoulis, N.; and Li, X. 2016. Meta structure: Computing relevance in large heterogeneous information networks. In *KDD*, 1595–1604.

Huang, J.; Nie, F.; and Huang, H. 2015. A new simplex sparse learning model to measure data similarity for clustering. In *IJCAI*, 3569–3575.

Kipf, T. N., and Welling, M. 2016. Variational graph auto-encoders. *arXiv preprint arXiv:1611.07308*.

Li, X.; Kao, B.; Zheng, Y.; and Huang, Z. 2016. On transductive classification in heterogeneous information networks. In *CIKM*, 811–820.

Li, X.; Wu, Y.; Ester, M.; Kao, B.; Wang, X.; and Zheng, Y. 2017. Semi-supervised clustering in attributed heterogeneous information networks. In *WWW*, 1621–1629.

Li, X.; Kao, B.; Luo, S.; and Ester, M. 2018. Rosc: Robust spectral clustering on multi-scale data. In *WWW*, 157–166.

Lin, F., and Cohen, W. W. 2010. Power iteration clustering. In *ICML*, 655–662.

Liu, J.; Wang, C.; Danilevsky, M.; and Han, J. 2013. Large-scale spectral clustering on graphs. In *IJCAI*, 1486–1492.

Long, B.; Zhang, Z. M.; Wu, X.; and Yu, P. S. 2006. Spectral clustering for multi-type relational data. In *ICML*, 585–592.

Marsden, A. 2013. Eigenvalues of the laplacian and their relationship to the connectedness of a graph. *University of Chicago, REU*.

Meila, M., and Shi, J. 2001. A random walks view of spectral segmentation.

Nadler, B.; Lafon, S.; Kevrekidis, I.; and Coifman, R. R. 2005. Diffusion maps, spectral clustering and eigenfunctions of fokker-planck operators. In *NIPS*, 955–962.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *NIPS*, 849–856.

Nie, F.; Wang, X.; Jordan, M. I.; and Huang, H. 2016. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, 1969–1976.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *KDD*, 701–710.

Sengupta, S., and Chen, Y. 2015. Spectral clustering in heterogeneous networks. *Statistica Sinica* 1081–1106.

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *TPAMI* 22(8):888–905.

Shi, Y.; Gui, H.; Zhu, Q.; Kaplan, L.; and Han, J. 2018. Aspem: Embedding learning by aspects in heterogeneous information networks. In *SDM*, 144–152.

Song, Y.; Chen, W.-Y.; Bai, H.; Lin, C.-J.; and Chang, E. Y. 2008. Parallel spectral clustering. In *ECML-PKDD*, 374–389.

Sun, Y.; Aggarwal, C. C.; and Han, J. 2012. Relation strength-aware clustering of heterogeneous information networks with incomplete attributes. *PVLDB* 5(5):394–405.

Sun, Y.; Han, J.; Yan, X.; Yu, P. S.; and Wu, T. 2011. Pathsim: Meta path-based top-k similarity search in heterogeneous information networks. *PVLDB* 4(11):992–1003.

Sun, Y.; Norick, B.; Han, J.; Yan, X.; Yu, P. S.; and Yu, X. 2012. Integrating meta-path selection with user-guided object clustering in heterogeneous information networks. In *KDD*, 1348–1356.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *WWW*, 1067–1077.

Tang, J.; Qu, M.; and Mei, Q. 2015. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, 1165–1174.

Tong, H.; Faloutsos, C.; and Pan, J.-Y. 2006. Fast random walk with restart and its applications. In *ICDM*, 613–622.

Yu, S. X., and Shi, J. 2003. Multiclass spectral clustering. In *ICCV*, 313–319.

Zelnik-Manor, L., and Perona, P. 2005. Self-tuning spectral clustering. In *NIPS*, 1601–1608.

Zhang, X.; Li, H.; Liang, W.; and Luo, J. 2016. Multi-type co-clustering of general heterogeneous information networks via nonnegative matrix tri-factorization. In *ICDM*, 1353–1358.

Zhou, Y., and Liu, L. 2013. Social influence based clustering of heterogeneous information networks. In *KDD*, 338–346.